Amendments to the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

Listing of the Claims:

1-13 (Cancelled)

14. **(Currently Amended)** A method for determining whether a field of a Java^{TM} object is a reference to another Java^{TM} object, said method comprising:

reading a class file associated with a Java^{TM} object;

identifying fields of said Java^{TM} object that are references;

allocating a reference identifier for said Java^{TM} object, wherein said reference identifier has one or more entries and each of said one or more entries correspond to a field of a Java^{TM} object, and wherein each of said one or more entries can be used to indicate whether a corresponding field of said Java^{TM} object is a reference to another Java^{TM} object;

initializing said reference identifier to indicate which fields of said Java^{TM} object are references;

identifying an internal class representation for a Java^{TM} object;

identifying a reference identifier in said internal class representation;

reading a portion of said reference identifier that represents said field of said Java^{TM} object; and

determining whether the value stored in said portion of said reference identifier is equal to a predetermined value.

15. (Original) A method as recited in claim 14, wherein said method is performed by a Java^{TM} virtual machine at runtime.

16. (Original) A method as recited in claim 14,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java^{TM} object.

17. (Original) A method as recited in claim 14, wherein said predetermined value can be 1 or zero.

18-20. (Cancelled)

21. (New) A method as recited in claim 16, wherein said predetermined value can be 1 or zero.

22. (New) A method as recited in claim 14, wherein said array of bytes is allocated and initialized during load time.

23. (New) A method as recited in claim 14, wherein said identifying of said internal class representation is performed at runtime.

24. (New) A computer readable medium including computer program code for determining whether a field of a Java^{TM} object is a reference to another Java^{TM} object, said computer readable medium comprising:

computer code for reading a class file associated with a Java^{TM} object;

computer code for identifying fields of said Java^{TM} object that are references;

computer code for allocating a reference identifier for said Java^{TM} object, wherein said reference identifier has one or more entries and each of said one or more entries correspond to a field of a Java^{TM} object, and wherein each of said one or more entries can be used to indicate whether a corresponding field of said Java^{TM} object is a reference to another Java^{TM} object;

computer code for initializing said reference identifier to indicate which fields of said Java^{TM} object are references;

computer code for identifying an internal class representation for a Java^{TM} object;

computer code for identifying a reference identifier in said internal class representation;

computer code for reading a portion of said reference identifier that represents said field of said Java^{TM} object; and

computer code for determining whether the value stored in said portion of said reference identifier is equal to a predetermined value.

25. (New) A computer readable medium as recited in claim 24, wherein said method is performed by a Java^{TM} virtual machine at runtime.

26. (New) A computer readable medium as recited in claim 24,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java^{TM} object.

27. (New) A computer readable medium as recited in claim 24, wherein said predetermined value can be 1 or zero.

28. (New) A computer readable medium as recited in claim 27, wherein said predetermined value can be 1 or zero.

29. (New) A computer readable medium as recited in claim 24, wherein said array of bytes is allocated and initialized during load time.

30. (New) A computer readable medium as recited in claim 24, wherein said identifying of said internal class representation is performed at runtime.

31. (New) A virtual machine for determining whether a field of a Java™ object is a reference to another Java™ object, wherein said virtual machine is capable of:

reading a class file associated with a Java™ object;

identifying fields of said Java™ object that are references;

allocating a reference Identifier for said Java™ object; wherein said reference identifier has one or more entries and each of said one or more entries correspond to a field of a Java™ object, and wherein each of said one or more entries can be used to indicate whether a corresponding field of said Java™ object is a reference to another Java™ object;

initializing said reference identifier to indicate which fields of said Java™ object are references;

identifying an internal class representation for a Java™ object;

identifying a reference identifier in said internal class representation;

reading a portion of said reference identifier that represents said field of said Java™ object; and

determining whether the value stored in said portion of said reference identifier is equal to a predetermined value.

32. (New) A virtual machine as recited in claim 31,

wherein said reference identifier is an array of bytes; and

wherein the size of said reference identifier is the same as the number of fields of said Java™ object.

33. (New) A virtual machine as recited in claim 31, wherein said predetermined value can be 1 or zero.

34. (New) A virtual machine as recited in claim 31, wherein said array of bytes is allocated and initialized during load time.

35. (New) A virtual machine as recited in claim 31, wherein said identifying of said internal class representation is performed at runtime.